

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Žiga Šinko

**RAZVOJ APLIKACIJE ZA OBRAČUN
STORITEV**

DIPLOMSKO DELO
NA UNIVERZITETNEM ŠTUDIJU

Mentor: doc. dr. Rok Rupnik

Ljubljana, 2013

Rezultati diplomskega dela so intelektualna lastnina avtorja in Fakultete za računalništvo in informatiko Univerze v Ljubljani. Za objavljane ali izkoriščanje rezultatov diplomskega dela je potrebno pisno soglasje avtorja, Fakultete za računalništvo in informatiko ter mentorja.



Št. naloge: 01964 / 2013
Datum: 18.10.2013

Univerza v Ljubljani, Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Kandidat: **ŽIGA ŠINKO**

Naslov: **RAZVOJ APLIKACIJE ZA OBRAČUN STORITEV**
THE DEVELOPMENT OF APPLICATION FOR BILLING OF SERVICES

Vrsta naloge: Diplomsko delo univerzitetnega študija

Tematika naloge:

Obračun storitev je v primeru, ko storitve zaračunavamo veliko strankam po različnih cenovnih modelih, lahko zelo kompleksen sistem. Njegovo kompleksnost praviloma povečujejo še različni urniki obračuna ter specifične zahteve strank. Zasnujte in izdelajte načrt za aplikacijo za obračun storitev elektronskega arhiviranja dokumentov. Pri tem upoštevajte vse zahteve naročnika in opredeljena poslovna pravila za naslednja področja: cenovni modeli, specifikacije storitev za obračun, urniki obračunov ter analize in pregledi obračuna. Aplikacijo razvijte na Microsoft platformi in pri tem uporabite tudi Windows Reporting Services.

Mentor:

doc. dr. Rok Rupnik



Dekan:

prof. dr. Nikolaj Zimic

IZJAVA O AVTORSTVU DIPLOMSKEGA DELA

Spodaj podpisani Žiga Šinko, z vpisno številko **63080152**, sem avtor diplomskega dela z naslovom:

Razvoj aplikacije za obračun storitev

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal samostojno pod mentorstvom doc. dr. Roka Rupnika
- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela
- soglašam z javno objavo elektronske oblike diplomskega dela v zbirki »Dela FRI«.

V Ljubljani, dne 23.10.2013

Podpis avtorja:

Zahvala

Zahvaljujem se mentorju, doc. dr. Roku Rupniku, za pomoč, vodenje in svetovanje pri izdelavi diplomske naloge.

Zahvalo izrekam tudi sodelavcem podjetja Mikrocop d.o.o., še posebej mentorju Klemnu Novaku, za pomoč in tehnično podporo pri izdelavi programske rešitve.

Posebno zahvalo namenjam svojim staršem in sestri za vso podporo in pomoč na poti do diplome.

Kazalo

<i>POVZETEK</i>	<i>1</i>
<i>ABSTRACT</i>	<i>1</i>
<i>SEZNAM UPORABLJENIH KRATIC IN SIMBOLOV</i>	<i>1</i>
<i>1 UVOD</i>	<i>1</i>
<i>2 UPORABLJENA ORODJA IN TEHNOLOGIJE</i>	<i>3</i>
2.1 Visual Studio 2012.....	3
2.2 Microsoft SQL Server.....	3
2.3 Visual C#, .NET Framework	3
2.4 ADO.NET Entity Framework	4
2.4.1 Linq to Entities.....	5
2.5 WPF.....	5
2.6 Caliburn.Micro.....	6
2.7 Windows Reporting Services	6
<i>3 PROBLEMSKA DOMENA</i>	<i>7</i>
3.1 Mikrocop	7
3.2 Storitve, ki se obračunavajo s pomočjo razvitega sistema	8
3.2.1 InDoc RMS.....	8
3.2.2 Brezpapirja.si.....	9
3.3 Obračunavanje storitev.....	10
3.3.1 Poizvedbe za pridobivanje podatkov o količini uporabe storitev	11
3.3.1.1. Master.....	12
3.3.1.2. Detail	12
<i>4 PROGRAMSKA REŠITEV</i>	<i>13</i>
4.1 Arhitektura sistema.....	13
4.2 Podatkovna baza	13
4.2.1 Šifranti.....	14

4.2.2	<i>Šifranti za generacijo obračunov</i>	<i>16</i>
4.2.3	<i>Podatkovno skladišče</i>	<i>21</i>
4.3	<i>Administracijski vmesnik</i>	<i>24</i>
4.3.1	<i>Delovanje.....</i>	<i>24</i>
4.3.1.1.	<i>Optimizacija poizvedb administracijskega vmesnika.....</i>	<i>25</i>
4.3.2	<i>Osnovni pogled aplikacije</i>	<i>25</i>
4.3.2.1.	<i>Navigacija.....</i>	<i>26</i>
4.3.3	<i>Dodajanje in urejanje šifrantov</i>	<i>27</i>
4.3.3.1.	<i>Validacija.....</i>	<i>28</i>
4.3.4	<i>Brisanje šifrantov.....</i>	<i>29</i>
4.3.5	<i>Uporaba preglednice.....</i>	<i>29</i>
4.3.5.1.	<i>Filtriranje</i>	<i>29</i>
4.3.5.2.	<i>Dodajanje več artiklov hkrati.....</i>	<i>29</i>
4.4	<i>Servisni modul.....</i>	<i>30</i>
4.4.1	<i>Izvajanje opravil.....</i>	<i>30</i>
4.4.2	<i>Postopek generiranja obračuna</i>	<i>31</i>
4.4.2.1.	<i>Testno generiranje obračuna.....</i>	<i>32</i>
4.5	<i>Windows reporting services.....</i>	<i>32</i>
4.6	<i>Uporaba programske rešitve</i>	<i>33</i>
4.7	<i>Nadaljnji razvoj</i>	<i>34</i>
5	<i>ZAKLJUČEK.....</i>	<i>37</i>
	<i>LITERATURA.....</i>	<i>39</i>

Povzetek

Diplomsko delo opisuje programsko rešitev za pripravo obračunov storitev v podjetju Mikrocop. S pomočjo razvitega sistema lahko v administracijskem vmesniku enostavno določimo specifikacije obračunov vseh strank in urnik generiranja teh obračunov. Te nastavitve upošteva servisni modul, ki se neprestano izvaja in generira obračune v skladu z določenim urnikom. Pripravljene obračune lahko kadarkoli pregledamo in z njimi manipuliramo s pomočjo Windows Reporting Services.

Uvodno poglavje v splošnem opisuje področje obračunavanja storitev.

Drugo poglavje na kratko opiše uporabljena orodja in tehnologije pri razvoju programske rešitve Mikrocop Billing System.

Tretje poglavje predstavi problemsko domeno, ki smo jo pokrili z razvitim sistemom. Najprej je v splošnem predstavljeno podjetje Mikrocop in storitve, ki jih ponuja. Nato sta podrobno opisani ključni storitvi, ki ju naročnikom obračunavamo z razvitim sistemom. To sta storitev za arhiviranje InDoc RMS in storitev brezpapirja.si. Zadnji razdelek tega poglavja obsega splošen pregled rešitve problemske domene – načina delovanja sistema za pripravo obračunov storitev.

Četrto poglavje obsega opis končnega produkta Mikrocop Billing System. Prvi sklop opisuje celotno arhitekturo razvitega sistema in način povezovanja med posameznimi deli. Nato so podrobneje opisani entitetni tipi in relacije med njimi v podatkovni bazi sistema. Tretji sklop je namenjen predstavitvi delovanja in principov uporabe administracijskega vmesnika. Četrti sklop opisuje delovanje servisnega modula za generiranje obračunov (polnjenje podatkovnega skladišča). Zadnji sklop opisovanja razvitega sistema je opis delovanja Windows Reporting Services in končnega rezultata – obračuna. Nato je opisan postopek uporabe Mikrocop Billing Systema od urejanja nastavitvev do prikaza končnega obračuna. Nazadnje pa so predstavljene ideje za izboljšanje sistema in možnosti nadaljnjega razvoja.

KLJUČNE BESEDE: Mikrocop, obračun, Windows Reporting Services, podatkovno skladišče, InDoc RMS, brezpapirja.si

Abstract

Diploma is a description of a programming solution for preparing billing invoices of Mikrocop services. Developed system can easily manage billing invoices specifications for all customers and set schedules for generating invoices. These settings are then used by the service module generating invoices on schedule. Generated invoices can be seen and managed with help of Windows Reporting Services technology.

First chapter describes services billing in general.

Second chapter shortly describes the tools and technologies used in developing Mikrocop Billing System.

Third chapter presents Mikrocop services that use the developed system for billing. The main two are an archiving solution InDoc RMS and brezpapirja.si. The last section of this chapter presents a general solution of the problem – description of Mikrocop Billing System process.

The fourth chapter is presents the finished system. The first section shows the architectural structure of the developed system and relationship between separate parts. Then the system's database entity types and connections between them are presented in detail. The third section is used to show administrative user interface process and usage principles. Fourth section describes the steps in generating billing invoices (filling data warehouse) in Mikrocop Billing System service module. The last section that describes the finished system is a description of Windows Reporting Services usage and the final product – invoice. Then a sample of usage from managing the settings to showing the final invoice is shown. Lastly we give some ideas for improving the Mikrocop Billing System.

KEYWORDS: Mikrocop, invoice, Windows Reporting Services, data warehouse, InDoc RMS, brezpapirja.si

Seznam uporabljenih kratic in simbolov

MVVM – Model View ViewModel;

SQL – Structured Query Language; strukturirani povpraševalni jezik za delo s podatkovnimi bazami

XML – Extensible Markup Language; razširljiv označevalni jezik, ki omogoča format za opisovanje strukturiranih podatkov in njihovo izmenjavo med omrežji

ID – Identifier; enolični identifikator

ADO.NET – ActiveX Data Object; zbirka programskih komponent za dostop do podatkov in podatkovnih servisov

LINQ – Language Integrated Query; komponenta Microsoft .NET ogrodja za poizvedovanje po podatkovnih strukturah v aplikaciji na način podoben poizvedovanju po podatkovnih bazah (SQL)

ORM – Object-Relational Mapping; programska tehnika za pretvorbo podatkov med nezdružljivimi sistemi v objektno orientiranih programerskih jezikih

WPF – Windows Presentation Foundation

XAML – Extensible Application Markup Language; deklarativni XML jezik za inicializacijo strukturiranih objektov

RDL – Report Definition Language; Microsoft standard za definiranje poročil

ISO – International Organization for Standardization

API – Application Programming Interface; programski vmesnik

WCF – Windows Communication Foundation; izvajalni sistem (ang. runtime) in zbirka API-jev v .NET ogrodju za razvijanje povezanih, servisno orientiranih aplikacij

CSV – Comma-separated Values; CSV datoteka vsebuje podatke, ki so v tekstu ločeni z določenim znakom (npr. »,«)

GUID – Globally unique identifier; globalni unikatni identifikator

1 Uvod

V svetu poznamo najrazličnejše načine obračunavanja storitev. Izbira pravega je v največji meri odvisna od narave storitve. Pri storitvah, ki jih posamezniki uporabljajo manj pogosto (npr. striženje las), je najbolj smotrno poravnati ceno storitve neposredno pred oz. po uporabi. Drugače velja za storitve z višjo frekvenco uporabe, ki jih je najenostavnejše obračunati v začetku ali koncu obračunskega obdobja (npr. mesec dni). Za te vrste storitev večina storitvenih podjetij, ki temeljijo na intervalnem obračunavanju (mobilni operaterji, ...), ponuja različne pakete uporabe. Na te pakete se naročniki vežejo že pred samo uporabo in v primeru prekoračitve omejitev paketa po višji tarifi poravnajo presežek.

Cena uporabe storitve je zlasti v poslovnem svetu mnogokrat odvisna od posebnih dogovorov s stranko. Tako imajo večji porabniki lahko bolj ugodno ceno posamezne enote uporabe storitve, saj ceno v večji meri predstavlja strošek razvoja in ne uporabe.

Način obračunavanja storitev v podjetju Mikrocop temelji na mesečnem obračunu uporabe storitev. Posebnost obračunavanja se kaže v načinu pridobivanja podatkov o uporabi storitev. Stranke imajo možnost namestitve programske opreme za izvajanje storitev v lastnih prostorih ali pa v varni sobi stavbe Mikrocop. Zato mora priprava mesečnih obračunov potekati enako za vse lokacije namestitve.

Mikrocop ima veliko naročnikov, ki uporabljajo več produktov na različnih lokacijah. Zato je bilo potrebno razviti celovit sistem, ki bi na enem mestu izdelal prilagodljive račune za vse stranke in njihove poljubne sisteme. Prav tako pa mora ustrezno upoštevati vse posebnosti pogodb uporabe z vsemi strankami.

Cilj projekta je bil, da bi določeni zaposleni podjetja Mikrocop lahko za vsako stranko posebej določil katere postavke se morajo obračunati ter čas generacije vsakega obračuna. Obračuni se morajo najprej vnesti v podatkovno skladišče, iz katerega se nato s spletno storitvijo pridobijo poročila, ki se v več oblikah lahko posredujejo strankam.

2 Uporabljena orodja in tehnologije

2.1 Visual Studio 2012

Visual Studio 2012 je razvojno orodje, ki ga je razvil Microsoft. Omogoča razvoj programskih rešitev za vse platforme, ki jih podpira korporacija Microsoft. Orodje ima vključenih več programskih jezikov (C/C++, VB.NET, C#, F#).

Izjemno dobro je realiziran urejevalnik kode, ki s tako imenovano funkcijo »IntelliSense« pospeši proces razvoja. Pomaga pri identifikaciji tipkarskih napak, označevanju sintakse, dopolnjevanju kode, hitremu prikazovanju opisov funkcij, ...

Napake v programski rešitvi lahko odpravimo s pomočjo vključenega razhroščevalnika (»Debugger«). Uporablja se lahko za aplikacije, ki so napisane v katerem koli od podprtih programskih jezikov. S pomočjo tako imenovanih »breakpoint«-ov, se lahko prekine izvajanje aplikacije in nadaljuje na željenem mestu oz. izvaja zapisano kodo vrstico po vrstico.

Visual Studio 2012 vsebuje tudi orodja za manipulacijo vseh datotek programske rešitve, raziskovalca povezav do podatkovnih baz, vizualni urejevalnik (»Designer«), ...

Za potrebe implementiranja programske rešitve Mikrocop.Billing, je v Visual Studio 2012 treba dodati paket Caliburn.Micro. To najbolj enostavno omogoča vtičnik Nuget, s katerim enostavno izberemo paket Caliburn.Micro na spletu in ga vključimo v projekt.

2.2 Microsoft SQL Server

Microsoft SQL Server je sistem za upravljanje z relacijskimi podatkovnimi bazami. Osnovni programski jezik za komunikacijo s strežnikom je SQL, natančneje T-SQL (Transact-SQL). T-SQL je implementacija standarda SQL-92, ki je standard ISO za SQL iz leta 1992.

Programsko rešitev Mikrocop.Billing smo razvijali in povezali na Microsoft SQL Server 2008 R2.

2.3 Visual C#, .NET Framework

C# je objektno naravnan programski jezik za razvoj najrazličnejših aplikacij v .NET Framework-u.

Visual C# je Microsoft implementacija C# jezika. Uporablja se v razvojnem okolju Visual Studio z vsemi funkcionalnostmi, ki jih omogoča to orodje.

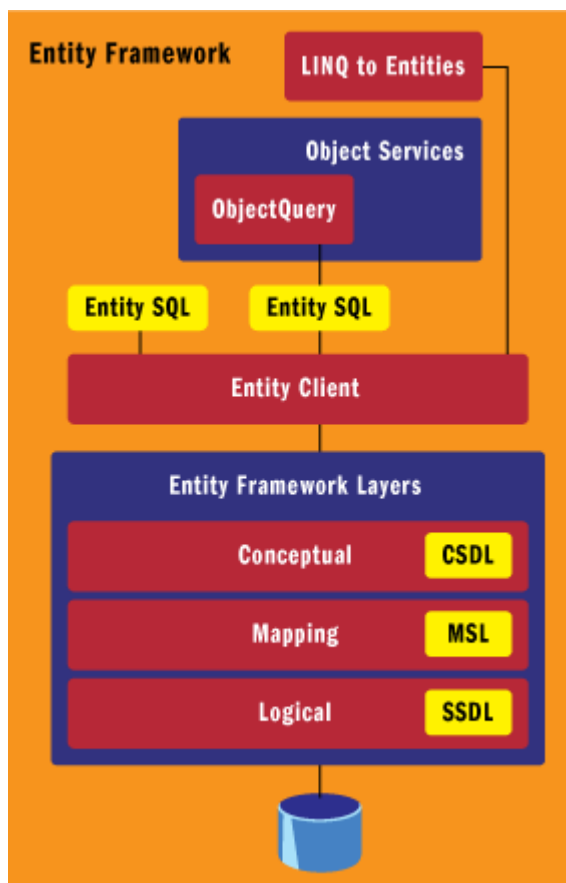
.NET Framework je knjižnica razredov, ki vključujejo mnogo dobro razdelanih uporabnih funkcionalnosti za pohitritev in poenostavitev razvoja. V programski rešitvi Mikrocop.Billing smo uporabili ogrodje Microsoft .NET Framework 4.0.

2.4 ADO.NET Entity Framework

Glavna naloga ADO.NET Entity Framework-a je zvišanje nivoja abstrakcije pri dostopanju do podatkov. V entitetno-relacijskem modelu (.edmx) je definiran konceptualni model (objekti in povezave med njimi v aplikaciji), logični model (objekti in povezave med njimi v podatkovnem viru) in mapiranje med tema dvema modeloma. Na sliki 1 je prikazana shema arhitekture entitetnega-relacijskega ogrodja. Tako nam ni več potrebno razvijati dodatne kode za izmenjavo podatkov med aplikacijo in ostalimi različnimi podatkovnimi viri. Aplikacije razvite na ADO.NET Entity Frameworkom so enostavno prilagodljive različnim podatkovnim virom oz. shemam.

Z izdelavo entitetno-relacijskega modela lahko vse operacije izmenjave podatkov implementiramo kar na ta model. Ogrodje ADO.NET Entity Framework pa bo poskrbelo za povezavo med našim konceptualnim modelom in dejanskimi podatki v podatkovnem viru (v našem primeru podatkovna baza Microsoft SQL Server).

Entitetno-relacijski model lahko generiramo iz obstoječe podatkovne baze ali pa najprej definiramo objekte znotraj aplikacije iz katerih se nato avtomatsko generira celotna shema podatkovne baze.



Slika 1 Diagram arhitekture ADO.NET Entity Framework-a

2.4.1 Linq to Entities

Kot nam pove že ime, nam Linq del omogoča lažje pridobivanje podatkov. Želeni rezultat lahko zapišemo le v obliki poizvedbe.

Linq to Entities je eden od novih Microsoftovih ORM produktov. ORM pomeni Object-Relational Mapping (objektno – relacijsko povezovanje). Gre za zbirko razredov, ki mapirajo objekte relacijske podatkovne baze z objekti v aplikaciji. Z Linq podporo omogoča pisanje poizvedb glede na objekte, zapisane v konceptualnem modelu.

2.5 WPF

WPF je predstavitevni sistem za razvoj najrazličnejših samostojnih oz. v brskalniku gostujočih aplikacij. Temelji na ločevanju vizualnega izgleda aplikacije (View – pogled) od implementacije delovanja (ViewModel – kontroler).

Izgled aplikacije je definiran v XAML deklarativnem »markup« programskem jeziku. XAML datoteke so v bistvu XML datoteke s končnico .xaml. V njej so z atributi definirana okna,

dialogi, strani, uporabniške kontrole, razne oblike in grafike. Funkcionalnosti ob proženju dogodkov uporabniškega vmesnika so določene v drugem razredu, ki je določen v atributu glavnega elementa.

Glavne funkcionalnosti aplikacije so zapisane v ViewModel (»Code-Behind«) razredu. Ta razred mora dedovati od elementa, na katerem je določen.

2.6 Caliburn.Micro

Caliburn.Micro je majhno ogrodje, idealno kot podpora za lažje in hitrejšo razvijanje MVVM aplikacij. Omogoča enostavno povezovanje ViewModela (kontrolerja) z View-om (pogledom), prikazovanju posameznih pogledov in pošiljanju sporočil med posameznimi ViewModeli. Povezava med pogledom in ViewModelom je realizirana samo z ustreznim poimenovanjem. Če imenu pogleda dodamo besedo »Model«, dobimo poimenovanje ViewModela za ta pogled. Na primer pogled AddCustomerView ima kontroler implementiran v razredu AddCustomerViewModel.

2.7 Windows Reporting Services

Reporting Services je strežniško osnovana programska rešitev, ki omogoča generiranje, upravljanje in pošiljanje poročil iz različnih podatkovnih virov.

Predloge poročil definiramo z RDL v obliki XML. Uporabniki lahko do poročil dostopajo direktno na strežniku, prek spletne aplikacije ali v kateri koli aplikaciji v katero smo vključili ASP.NET (ali Windows Forms) ReportViewer kontrolo. Spletna aplikacija uporabnikom omogoča pregled poročil, naročevanje na ponavljajoča poročila, urejanje določenih nastavitev predlog poročil, ...

3 Problemska domena

3.1 Mikrocop

Mikrocop je slovensko podjetje, ki se ukvarja z upravljanjem dokumentov, varno dolgoročno hrambo in infrastrukturnimi storitvami. Kvaliteta ponujenih storitev je dokazana z uspešno potrjenimi notranjimi pravili pri Arhivu Republike Slovenije (od decembra 2007), akreditacijo programske opreme (od decembra 2008) in vpeljavo standarda za vodenje varovanja informacij ISO/IEC 27001 (od oktobra 2012).

Kljub vedno večjemu deležu elektronskega poslovanja je večina dokumentov v organizacijah še vedno v papirni obliki. Zato je zaradi zagotavljanja hitrega dostopa in dolgoročne hrambe potrebno pretvoriti papirne dokumente v elektronsko obliko. Zajem dokumentov se izvaja v pritličnih prostorih podjetja Mikrocop. Tam skeneristi v skladu s potrjenimi notranjimi pravili zajemajo gradivo v digitalno obliko oz. na mikrofilm.

Poleg zajema dokumentov iz papirne v elektronsko obliko se ti tudi pretvorijo v ustrezno obliko za dolgoročno hrambo. Ko so dokumenti enkrat v elektronski obliki, se nad njimi lahko izvede indeksacija. To je postopek strojnega in redkeje človeškega razbiranja podatkov iz slik dokumentov. Z indeksacijo se močno znižajo stroški vnosa podatkov v informacijski sistem, skrajša čas za procesiranje dokumentov in poveča kakovost vhodnih podatkov.

Sistem brezpapirja.si omogoča elektronsko upravljanje poslovnih procesov. Elektronski dokumenti v sistemu Brezpapirja.si potujejo v elektronski obliki in avtomatizirano v skladu z določenimi poslovnimi pravili. Uporabniki imajo celovit pregled nad vsemi fazami procesa. Ob zaključku poslovnega procesa se dokumenti varno shranijo v arhivskem sistemu Arhiviraj.si.

Dolgoročna hramba dokumentov je realizirana v sistemu InDoc RMS. Spletne storitve Arhiviraj.si omogočajo varno dolgoročno elektronsko hrambo vseh dokumentov (papirni pretvorjeni v elektronsko obliko in izvorno elektronski) na enem mestu. Poleg dolgoročne hrambe elektronskih dokumentov podjetje Mikrocop nudi tudi najem arhivskega prostora in storitve upravljanja z arhivi v papirni, mikrofilmski ali drugih podatkovnih nosilcih.

Zgradba podjetja Mikrocop je bila načrtovana z velikim poudarkom na zagotavljanju varne infrastrukture za hrambo občutljivih podatkov in neprestane razpoložljivosti. Varnost v poslovni zgradbi zagotavlja fizično in tehnično varovanje, video nadzor, lastna energetska oskrba, stalen nadzor klimatskih pogojev, protipožarna zaščita,... Poleg tega so strežniški sistemi nameščeni v varni sistemski sobi Lampertz. Ta vsebuje vrsto zaščitnih mehanizmov,

da lahko uspešno kljubuje vsem vrstam možnih napadov na vsebino sobe. Nesporno varnost Lampertz sistemskih sob potrjuje certifikat ECB – S (European Certification Body). Vstop vanjo imajo le pooblašчени zaposleni oz. potrjeni obiskovalci v obveznem spremstvu pooblaščenih. V teh varnih prostorih lahko stranke najamejo virtualne / fizične strežnike ali pa najamejo prostor v varni sobi za postavitve lastne opreme. Hkrati pa imajo stranke možnost najema ali nakupa prostorov v kletnih etažah poslovnega objekta Mikrocop.

3.2 Storitve, ki se obračunavajo s pomočjo razvitega sistema

S programsko rešitvijo Mikrocop Billing System se lahko obračunavajo storitve, ki shranjujejo ključne podatke v podatkovno bazo oz. vanjo vpisujejo informacije o količini uporabe. Trenutno (2013) so v te storitve povezane z arhiviranjem dokumentov (storitev Arhiviraj.si) in sistemom za upravljanje delovnih tokov (Brezpapirja.si).

3.2.1 InDoc RMS

Sistem za varno in dolgoročno hrambo gradiva. Primeren je tako za majhne kot velike količine gradiva.

Struktura sistema vključuje paleto servisov in komponent. Administracijski vmesnik za konfiguriranje vseh nastavitev za arhiviranje je namizna aplikacija. Za pregledovanje arhiviranih dokumentov in osnovno manipulacijo z njimi skrbi spletni vmesnik Arhiviraj.si. Poleg teh dveh komponent je v sistemu še več različnih servisov za izvajanje opravil kot so uvoz dokumentov, pripravo seznamov dokumentov, podpisovanje in kriptiranje dokumentov, ... Spletni servis InDoc RMS nudi integracijo arhiviranja v zunanje sisteme (npr. Brezpapirja.si). Vse komponente imajo vgrajeno zaščito dostopa, ki preverja uporabniške pravice dostopa (nastavimo jih v administracijskem vmesniku) in omogoča več načinov prijave (certifikat, uporabniško ime in geslo, aktivni imenik – Windows prijava).

V jedru sistema je najpomembnejša entiteta dokument. Dokument je vsak zapis, ki je opremljen z metapodatki. Aplikacija za vsak dokument omogoča hranjenje njegovih splošnih podatkov, metapodatkov, elementov hrambe, določanje urnikov shranjevanja na določeno pot shrambe in možnost digitalnega podpisovanja ter enkripcije (zagotavljanje varne dolgoročne hrambe).

Dokumenti se v sistem lahko uvozijo na več načinov. Največkrat uporabljen način je uvoz z servisnim modulom, ki v sistem uvozi vse dokumente na specificirani lokaciji v skladu z

pravili razčlenbe¹. Poleg servisnega uvoza je na voljo tudi čarovnik za uvoz dokumentov, posamične dokumente pa lahko uvozimo tudi prek spletnega vmesnika Arhiviraj.si. Zunanji sistemi pa imajo za uvoz na voljo funkcije spletnega servisa InDoc RMS.

3.2.2 Brezpapirja.si

Brepapirja.si je enoten portal za vodenje vseh procesov v podjetju. Izvajanje poslovnih procesov temelji na toku dokumentov, ki prihajajo v podjetje oz. v njem nastajajo. Pomembno je, da se vsi dokumenti obdelujejo enako, neodvisno od izvora dokumenta (papirni ali elektronski izvirnik). Ko elektronski dokumenti vstopijo v sistem, skozi celoten proces potujejo v elektronski obliki in avtomatizirano, uporabniki pa imajo nad vsemi fazami procesa celovit pregled. Tako zagotovimo, da vsak dokument skozi poslovni proces v elektronski obliki prepotuje najkrajšo možno pot, od sprejemne pisarne do zakonsko skladnega sistema elektronske hrambe.

Dokumenti se po zaključeni poti skozi poslovni proces varno hranijo v informacijskem sistemu InDoc RMS ali storitvi v oblaku arhiviraj.si. Dostop do elektronskih dokumentov shranjenih v varni dolgoročni hrambi je mogoč neposredno iz Brezpapirja.si.

V osnovi je sistem sestavljen iz dveh delov. Brezpapirja.si del je bil v celoti razvit v podjetju Mikrocop. Drugi del pa predstavlja mehanizem za vodenje delovnih tokov Skelta BPM.Net.

Brepapirja.si del vključuje spletni uporabniški in administrativni vmesnik. V spletni vmesnik se uporabnik lahko prijavi s certifikatom, uporabniškim imenom in geslom ali prek aktivnega imenika (Active Directory – Windows prijava). Spletni vmesnik omogoča oddajanje novih zahtevkov, opravljanje opravil, nastavljanje nadomestnega uporabnika, zgodovinski pregled oddanih zahtevkov, iskanje zahtevkov, ... S spletnim administracijskim vmesnikom lahko nastavimo vse podatke o organizacijah in njihovih uporabnikih, definicije poslovnih procesov, različne šifrante za lažje izpolnjevanje zahtevkov, ... Uporabniki imajo nastavljene pravice dostopa do procesov in urejanja nastavitvev.

Poleg spletnega vmesnika sistem Brezpapirja.si vključuje še integracijski (spletni) vmesnik za proženje procesov zunanjih aplikacij. Za prevzem elektronskih sporočil in proženje procesov se ozadju izvaja servisni modul.

¹ Pravila razčlenbe so pravila, na podlagi katerih se iz uvožene datoteke izlušči metapodatke za izbrani dokument. Z njimi za vsak metapodatek določimo, kje v uvoženi datoteki se nahaja.

Skelta BPM.Net je robustno orodje za upravljanje poslovnih procesov na Microsoft platformi, ki omogoča učinkovito modeliranje in izvajanje poslovnih procesov, ki jih je možno povezovati z zunanjimi sistemi s pomočjo .NET in WCF tehnologije. Modeliranje in administracija poslovnih procesov je omogočena s spletno administracijsko konzolo (Skelta Enterprise Console).

3.3 Obračunavanje storitev

Končni produkt Mikrocop Billing System mora biti obračun za izbrano stranko v določenem obdobju. Sam prikaz obračuna je enostavno realiziran z Windows Reporting Services. Ta tehnologija črpa podatke iz podatkovne baze. Zato je potrebno zgraditi podatkovno skladišče, ki bo hranilo podatke o obračunih s podrobnostmi za vse naročnike storitev.

Ker opisane storitve kot podatkovni vir uporabljajo podatkovno bazo, lahko za vse obstaja enoten sistem obračunavanja storitev. Podatke o uporabi storitev posameznih strank pridobimo s pomočjo različnih poizvedb po podatkovnih bazah storitev. Za vsako od možnih postavk obračuna je potrebno imeti pripravljeno poizvedbo, ki jo lahko posodobimo z identifikatorjem stranke in obračunskim obdobjem.

Podatkovne baze opisanih storitev so lahko nameščene vsaka na svojem strežniku. Ali na lokaciji podjetja Mikrocop ali pa gostujejo na strežniški opremi strank. Zato je potrebno za vsako stranko hraniti podatke o priklopu na podatkovno bazo (»Connection String«²) za vsako storitev, ki bo na obračunu te stranke. Te informacije pa se uporabijo kot mesto za izvedbo ustrezne poizvedbe.

Ključna funkcionalnost Mikrocop Billing Systema je polnjenje podatkovnega skladišča obračunov s podrobnostmi. Ker se računi strankam pošiljajo enkrat na mesec, se lahko vnašanje obračunov v podatkovno skladišče izvede sredi noči in tako s poizvedovanjem po podatkovni bazi ne moti rednega izvajanja storitev. S podatki, ki jih vrnejo poizvedbe lahko nato s cenikom določimo ceno postavke in jo vnesemo v račun podatkovnega skladišča.

Večina strank ima individualen cenik uporabe storitev. Zato je bilo potrebno v Mikrocop Billing System vključiti cenike za vse stranke in način cenoitve posameznih postavk.

Cena posameznih postavk uporabe je lahko določena na dva načina:

- paketno – enotna cena uporabe brez sprememb pri večjih količinah uporabe

² Connection String je tekstovni niz, ki specificira informacije o podatkovnem viru in načinu povezave nanj.

- lestvica – cena postavke je odvisna od količine uporabe. Primer:
 - Uporabniki od 1 do 100 – 10€
 - Uporabniki od 101 do 250 – 7€
 - Uporabniki nad 250 – 4€

Ker imajo stranke med seboj različen nabor uporabljenih storitev in postavk storitev, smo v Mikrocop Billing System vključili specifikacije obračunov. Tako lahko za posamezno stranko točno določimo katere postavke uporabe storitev morajo biti na računu.

Te specifikacije obračunov nato dodamo v opravila, ki jim določimo datum in čas izvedbe. Ob točno določeni uri se bodo tako v podatkovno skladišče vnesli obračuni s postavkami kot so zapisane v specifikacijah.

Celotno urejanje nastavitev delovanja Mikrocop Billing Systema je omogočeno z administracijskim vmesnikom. Ta vsebuje mnogo funkcionalnosti za lažjo administracijo in testiranje nastavitev.

3.3.1 Poizvedbe za pridobivanje podatkov o količini uporabe storitev

Pridobivanje podatkov o količini uporabe storitev lahko izvajamo s pomočjo poizvedb po podatkovnih bazah storitev. V poizvedbah lahko uporabimo več rezerviranih besed, ki se med izvajanjem servisnega modula nadomestijo z ustreznimi vrednostmi iz nastavitev v administracijskem vmesniku. Rezervirano besedo zapišemo v obliki »@« in imenom stolpca v podatkovni bazi (npr. »@vcSchema«). Na voljo so naslednje opcije:

- @vcSchema – shema podatkovne baze po kateri poizvedujemo
- @vcGlobalSchema – globalna shema podatkovne baze po kateri poizvedujemo
- @nInstanceID – identifikator instance v naročnikovem sistemu
- @vcInstanceName – ime instance v naročnikovem sistemu
- @uidInstanceGUID – GUID (globalni identifikator)
- @vcInstanceID - identifikator instance v naročnikovem sistemu (tekstovna oblika)
- @dtBillingPeriodStartDate – začetek obdobja obračuna
- @dtBillingPeriodEndDate – konec obdobja obračuna

Poizvedbe lahko podajo rezultate o uporabi storitev na dva načina. Na voljo sta dva tipa: »master« in »detail«.

3.3.1.1. Master

Master tip predstavlja agregiran rezultat poizvedbe. To pomeni, da nam poizvedba vrne le naziv vrste uporabe (npr. ime vrste dokumentov, ki so bili uvoženi) in količino uporabe te vrste (npr. količina uvoženih dokumentov u to vrsto dokumentov).

Primer »master« poizvedbe vseh paketno podpisanih in časovno žigosanih dokumentov v sistemu InDoc RMS

```
select      dt.vcName as Data,      Count(d.nDocumentID) as Count
from rms.CLS_DocumentTypes dt WITH(NOLOCK)
inner join rms.ARC_Documents d WITH(NOLOCK) on (d.nDocumentTypeID =
dt.nDocumentTypeID)
where      cast(dt.xmlTimestampSchedule as
xml).value(' (/DocumentTypeSignSchedule/DocumentCollectionDocumentsMax) [1]
', 'int') <> 1
and d.dtPostDate between @dtBillingPeriodStartDate and
@dtBillingPeriodEndDate

group by dt.vcName
```

3.3.1.2. Detail

Detail tip predstavlja neagreriran tip poizvedbe. Poizvedba nam vrne vse podrobnosti uporabe (naziv enote uporabe, npr. Žiga Šinko -aktivni uporabnik InDoc RMS) s količino, ki je navadno ena.

Primer »detail« poizvedbe vseh aktivnih uporabnikov v sistemu InDoc RMS

```
select vcFullName as Data, 1 as Count
from rms.USR_Users WITH(NOLOCK)
where      nOrganizationID = @nInstanceID
and nUserID not in (select distinct nUserID
from rms.USR_UserRoles WITH(NOLOCK)
where nRoleID in (select nRoleID
from rms.USR_Roles WITH(NOLOCK)
where vcRoleName like 'Internal user'))
and (bitDisabled = 0 or bitDisabled is null)
and (dtDeleted is null)
and (bitPassword = 1 OR bitCertificate = 1 OR bitWindows = 1)
```

4 Programska rešitev

Programska rešitev Mikrocop Billing System je zasnovana iz več delov. V tem poglavju so najprej opisani posamezni deli sistema in povezovanje med njimi. Nato je podrobneje opisana uporabljena podatkovna baza, administracijski vmesnik, servisni modul in sistem za generiranje poročil Windows Reporting Services. Na koncu pa so predstavljene še možnosti za nadaljnji razvoj sistema.

4.1 Arhitektura sistema

Mikrocop billing system je razdeljen na tri ločene sisteme:

- Administracijski vmesnik
- Servisni modul
- Microsoft Reporting Services

Z administracijskem vmesnikom se nastavijo vrednosti vseh šifrantov (podatki o strankah, njihovih sistemih, ceniki,...) in šifrantov za generacijo obračunov (specifikacije obračunov, poizvedbe, vrste postavk, opravila in urnik izvedbe opravil,...).

Obračuni se generirajo v Microsoft Windows servisnem modulu v skladu s pravili, ki se določijo v administracijskim vmesnikom. Servisni modul se neprestano izvaja in ko je čas za izvedbo opravila (določeno z urnikom) v podatkovno skladišče shrani podatke vseh obračunov, ki so vezani na to opravilo.

Obračuni se v obliki poročila prikažejo s pomočjo Windows Reporting Services. Filtriranje prikazanih obračunov iz podatkovnega skladišča se izvaja z izbiro skupine naročnikov ali samega naročnika in obdobja. Ta poročila je mogoče izvoziti v več oblik (XML, CSV, PDF, Excel,...).

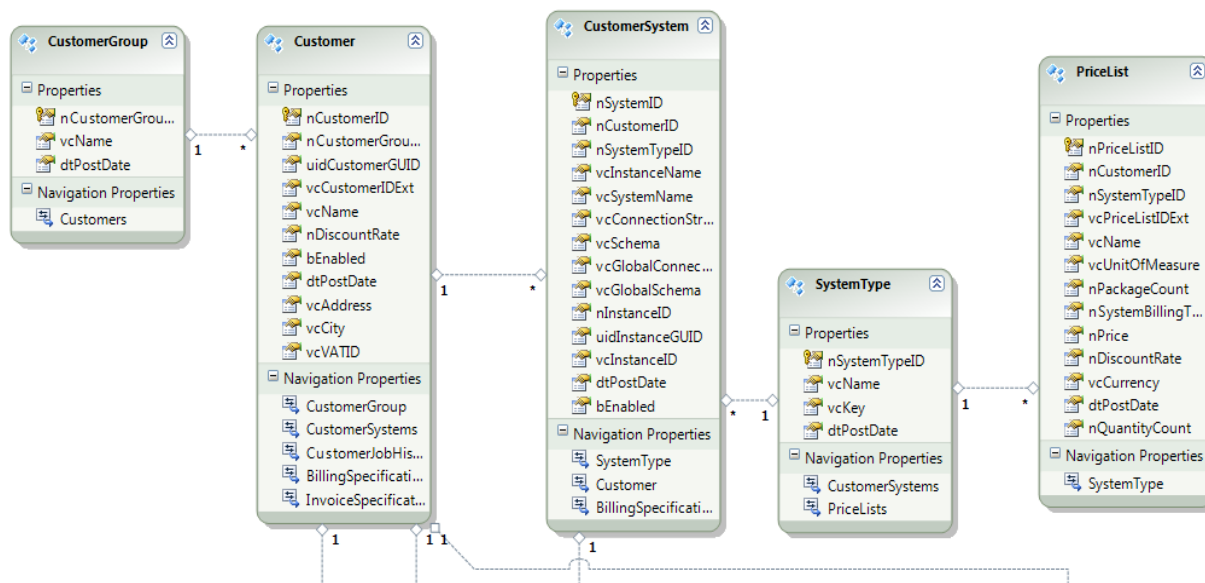
4.2 Podatkovna baza

Vsi trije deli sistema uporabljajo za delovanje isto podatkovno bazo. Uporablja tehnologijo Microsoft SQL Server 2008 R2. Pri razvoju programske rešitve smo uporabili pristop »Database first approach«. To pomeni, da je najprej potrebno izdelati model podatkovne baze iz katerega lahko enostavno generiramo programske objekte.

Model podatkovne baze vsebuje več entitetnih tipov, ki jih lahko razdelimo v štiri logične enote.

4.2.1 Šifranti

V šifrante shranjujemo informacije o naročnikih, njihovih sistemih in cenike.



Slika 2 Podatkovni model šifrantov

Tabela **Customer** hrani podatke o naročnikih.

Customer	Tip	Opis polja
nCustomerID	Int	Identifikator naročnika
nCustomerGroupID	Int	Identifikator skupine naročnikov
uidCustomerGUID	Uniqueidentifier	Globalni identifikator postavke
vcCustomerIDExt	nvarchar(100)	Zunanji ID naročnika
vcName	varchar(500)	Ime naročnika
nDiscountRate	decimal(16, 9)	Popust
bEnabled	Int	Zastavica ali je naročnik omogočen
dtPostDate	Datetime	Datum zapisa v podatkovno bazo
vcAddress	varchar(500)	Naslov naročnika
vcCity	varchar(500)	Mesto naročnika
vcVATID	varchar(50)	Davčna številka

V tabelo **CustomerGroup** se shranjujejo vse skupine naročnikov. Vsak naročnik mora biti pripadnik točno ene skupine.

CustomerGroup	Tip	Opis polja
nCustomerGroupID	int	Identifikator skupine naročnikov
vcName	varchar(500)	Ime skupine naročnikov
dtPostDate	datetime	Datum zapisa v podatkovno bazo

Tabela **CustomerSystem** hrani podatke o naročnikovih sistemih. Ti vključujejo tudi povezovalne nize na podatkovno bazo, ki se uporabijo v servisnem modulu pri odpiranju povezave do podatkovne baze nad katero se morajo izvesti poizvedbe. Poleg tega s to tabelo poizvedbi določi za katero instanco gre.

CustomerSystem	Tip	Opis polja
nSystemID	int	Identifikator naročnikovega sistema
nCustomerID	int	Identifikator naročnika
nSystemTypeID	int	Identifikator tipa sistema
vcInstanceName	varchar(500)	Ime instance
vcSystemName	varchar(500)	Ime sistema
vcConnectionString	varchar(MAX)	Povezovalni niz na podatkovno bazo
vcSchema	varchar(MAX)	Shema podatkovne baze
vcGlobalConnectionString	varchar(MAX)	Globalni povezovalni niz na podatkovno bazo
vcGlobalSchema	varchar(MAX)	Globalna shema podatkovne baze
nInstanceID	int	Številka instance
uidInstanceGUID	uniqueidentifier	Globalni identifikator instance
vcInstanceID	varchar(500)	Identifikator instance
dtPostDate	datetime	Datum zapisa v podatkovno bazo
bEnabled	int	Zastavica ali je naročnikov sistem omogočen

Tabela **PriceList** predstavlja cenik uporabe Mikrocop storitev. Posamezen artikel je lahko vezan na določenega naročnika ali pa ne – je splošen. Artikel ima vpisano ceno, ki velja za količino (nPackageCount) v merskih enotah (vcUnitOfMeasure). S poljem nQuantityCount je definirano za koliko enot artikla je zapisana cena.

PriceList	Tip	Opis polja
nPriceListID	int	Identifikator artikla

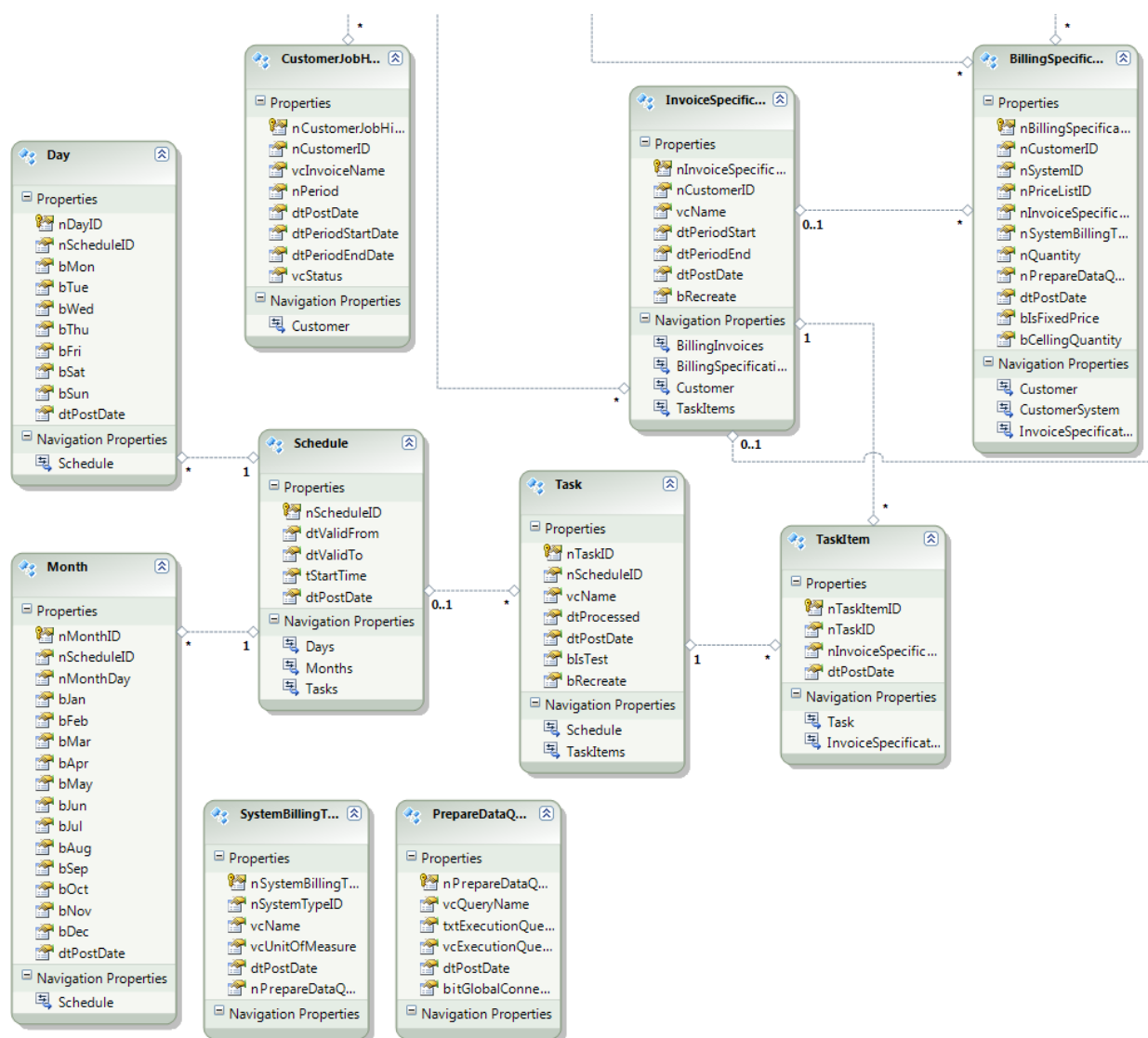
nCustomerID	int	Identifikator naročnika
nSystemTypeID	int	Identifikator tipa sistema
vcPriceListIDExt	varchar(100)	Zunanji ID artikla
vcName	varchar(500)	Ime artikla
vcUnitOfMeasure	varchar(100)	Merska enota
nPackageCount	decimal(38, 0)	Količina v artiklu
nSystemBillingTypeID	Int	Identifikator postavke
nPrice	decimal(16, 9)	Cena artikla
nDiscountRate	decimal(16, 9)	Popust
vcCurrency	varchar(100)	Valuta
dtPostDate	Datetime	Datum zapisa v podatkovno bazo
nQuantityCount	decimal(19, 6)	Količina artiklov na ceno

Tabela **SystemType** hrani vse tipe sistemov (produktov), ki jih ponuja Mikrocop.

SystemType	Tip	Opis polja
nSystemTypeID	Int	Identifikator tipa sistema
vcName	varchar(500)	Ime tipa sistema
vcKey	varchar(500)	Ključ tipa sistema
dtPostDate	Datetime	Datum zapisa v podatkovno bazo

4.2.2 Šifranti za generacijo obračunov

V šifrante za generacijo obračunov shranjujemo specifikacije na podlagi katerih se generirajo obračuni. Opravilu (**Task**) se nastavi urnik izvajanja (**Schedule** -> **Day**, **Month**). Na opravilo so vezane specifikacije računa (**InvoiceSpecification**), ki imajo več postavk (**BillingSpecification**) z določenim artiklom (**PriceList**) in opsijsko poizvedbo (**PrepareDataQuery**) ali lestvico (**SystemBillingType**). Status uspešnosti generacije se shranjuje v tabelo **CustomerJobHistory**.



Slika 3 Podatkovni model šifrantov za generiranje obračunov

V tabeli **Task** so shranjena vsa opravila. Na posamezno opravilo je vezano več specifikacij računov (**BillingSpecification**) in natanko en urnik (**Schedule**). Opravilo hrani datum in čas zadnje izvedbe, zato, da se v servisnem modulu izvede le enkrat v intervalu šest ur od časa določenega v urniku.

Task	Tip	Opis polja
nTaskID	int	Identifikator opravila
nScheduleID	int	Identifikator urnika
vcName	varchar(500)	Ime opravila
dtProcessed	datetime	Čas in datum začetka izvajanja opravila
dtPostDate	datetime	Datum zapisa v podatkovno bazo

bIsTest	int	Zastavica ali je obračun testni
bRecreate	Int	Zastavica ali se obračuni ponovno generirajo

Tabela **TaskItem** omogoča povezavo mnogo – mnogo med opravili in specifikacijami računov.

TaskItem	Tip	Opis polja
nTaskItemID	Int	Identifikator enote v opravilu
nTaskID	Int	Identifikator opravila
nInvoiceSpecificationID	Int	Identifikator specifikacije obračuna
dtPostDate	Datetime	Datum zapisa v podatkovno bazo

Schedule je urnik izvajanja opravila. Opravilo se bo izvedlo na čas v dnevu (tStartTime) na dneve v tednu in mesece, ki so določeni v povezanih tabelah Day in Month. Urnik mora imeti določeno obdobje veljavnosti.

Schedule	Tip	Opis polja
nScheduleID	Int	Identifikator urnika
dtValidFrom	Datetime	Datum začetka veljavnosti urnika
dtValidTo	Datetime	Datum konca veljavnosti urnika
tStartTime	time(7)	Časa začetka izvajanja
dtPostDate	Datetime	Datum zapisa v podatkovno bazo

InvoiceSpecification tabela predstavlja specifikacijo obračuna. Vezana je na naročnika in vključuje več postavk (BillingSpecification).

InvoiceSpecification	Tip	Opis polja
nInvoiceSpecificationID	Int	Identifikator specifikacije obračuna
nCustomerID	Int	Identifikator naročnika
vcName	varchar(500)	Ime obračuna
dtPeriodStart	Datetime	Začetni datum obdobja
dtPeriodEnd	Datetime	Končni datum obdobja
dtPostDate	Datetime	Datum zapisa v podatkovno bazo
bRecreate	Int	Zastavica ali se obračun ponovno generira

Tabela **BillingSpecification** definira postavke na obračunih. Vezana je na specifikacijo obračuna in na naročnikov sistem. Postavka je lahko fiksna ali vezana na lestvico. Če je fiksna, je določen artikel in opcijsko poizvedba – če želimo zapisati podrobnosti uporabe. Če pa je vezana na lestvico, je določena specifikacija postavke oz. lestvica s poljem nSystemBillingTypeID.

BillingSpecification	Tip	Opis polja
nBillingSpecificationID	int	Identifikator postavke
nCustomerID	int	Identifikator naročnika
nSystemID	int	Identifikator sistema
nPriceListID	int	Identifikator artikla
nInvoiceSpecificationID	int	Identifikator specifikacije obračuna
nSystemBillingTypeID	int	Identifikator lestvice
nQuantity	decimal(16, 9)	Količina
nPrepareDataQueryID	int	Identifikator poizvedbe
dtPostDate	datetime	Datum zapisa v podatkovno bazo
bIsFixedPrice	int	Zastavica fiksne cene
bCellingQuantity	int	Zastavica zaokroževanja količine navzgor

Tabela **CustomerJobHistory** opravlja funkcijo beleženja statusov generiranja obračunov v servisnem modulu. Vanjo se shrani uspešnost oz. napake ter podatki o obračunu za katerega je nastal status.

CustomerJobHistory	Tip	Opis polja
nCustomerJobHistoryID	int	Identifikator statusa generacije
nCustomerID	int	Identifikator naročnika
vcInvoiceName	varchar(500)	Naziv obračuna
nPeriod	int	Obdobje
dtPostDate	datetime	Datum zapisa v podatkovno bazo
dtPeriodStartDate	datetime	Začetni datum obdobja
dtPeriodEndDate	datetime	Končni datum obdobja
vcStatus	varchar(500)	Status generacije obračuna

Tabela **Day** je vezana na urnik in specificira na katere dneve v tednu se bo opravilo izvajalo.

Day	Tip	Opis polja
nDayID	int	Identifikator dneva
nScheduleID	int	Identifikator urnika
bMon	int	Zastavica ali se izvaja v ponedeljek
bTue	int	Zastavica ali se izvaja v torek
bWed	int	Zastavica ali se izvaja v sredo
bThu	int	Zastavica ali se izvaja v četrtek
bFri	int	Zastavica ali se izvaja v petek
bSat	Int	Zastavica ali se izvaja v soboto
bSun	Int	Zastavica ali se izvaja v nedeljo
dtPostDate	Datetime	Datum zapisa v podatkovno bazo

Tabela **Month** je vezana na urnik in določa na katere mesece v letu se bo opravilo izvajalo.

Month	Tip	Opis polja
nMonthID	Int	Identifikator meseca
nScheduleID	Int	Identifikator urnika
nMonthDay	Int	Dan v mesecu
bJan	Int	Zastavica ali se izvaja v januarju
bFeb	Int	Zastavica ali se izvaja v februarju
bMar	Int	Zastavica ali se izvaja v marcu
bApr	Int	Zastavica ali se izvaja v aprilu
bMay	Int	Zastavica ali se izvaja v maju
bJun	Int	Zastavica ali se izvaja v juniju
bJul	Int	Zastavica ali se izvaja v juliju
bAug	Int	Zastavica ali se izvaja v avgustu
bSep	Int	Zastavica ali se izvaja v septembru
bOct	Int	Zastavica ali se izvaja v oktobru
bNov	Int	Zastavica ali se izvaja v novembru
bDec	Int	Zastavica ali se izvaja v decembru
dtPostDate	Datetime	Datum zapisa v podatkovno bazo

Tabela **PrepareDataQuery** hrani poizvedbe, s katerimi ugotavljamo količino uporabe določene storitve. Za katero storitev gre je zapisano v tabeli **SystemBillingType**, ki je vezana na tabelo poizvedb.

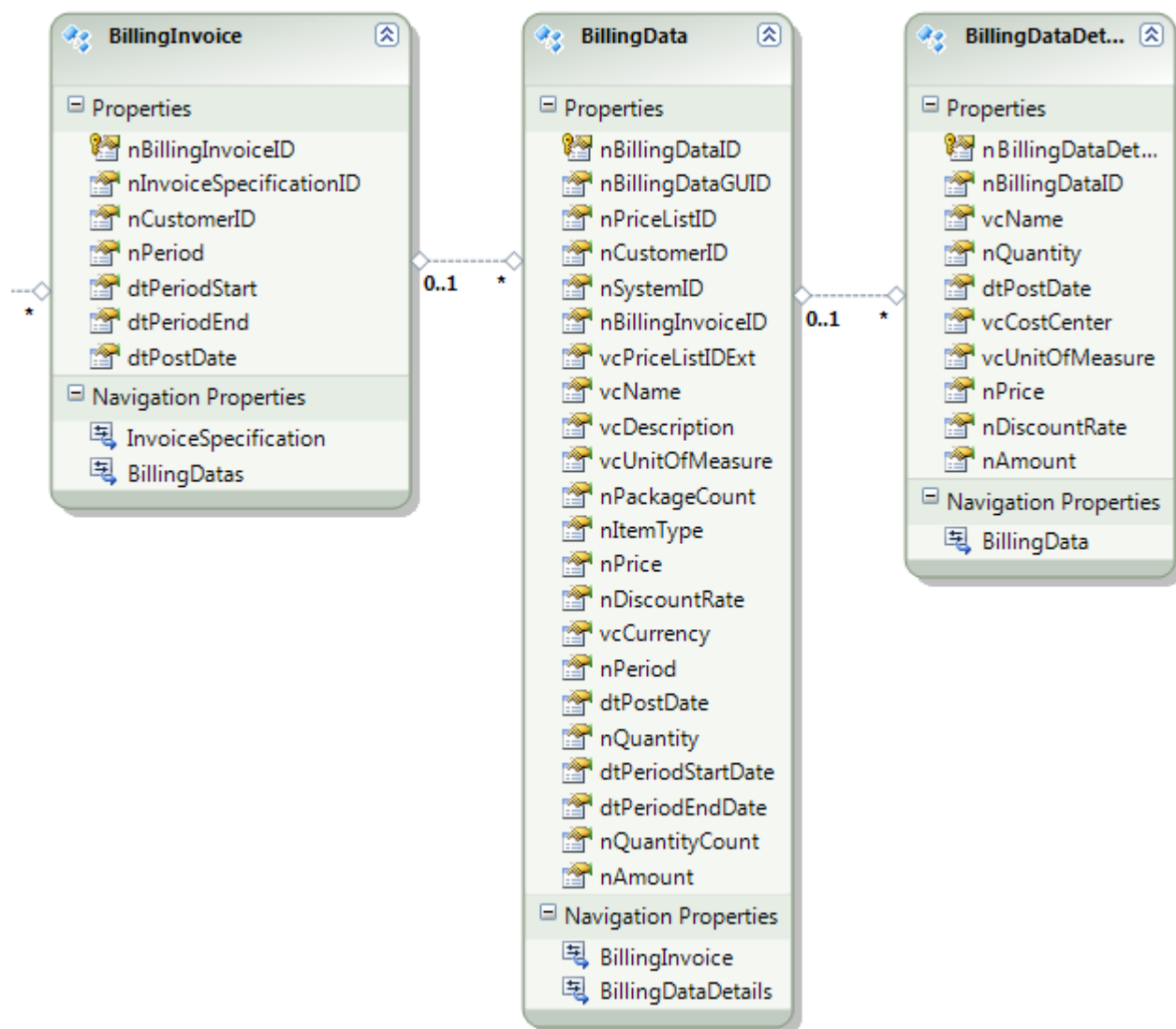
PrepareDataQuery	Tip	Opis polja
nPrepareDataQueryID	Int	Identifikator poizvedbe
vcQueryName	varchar(200)	Ime poizvedbe
txtExecutionQuery	Text	Tekst poizvedbe
vcExecutionQueryType	varchar(100)	Tip poizvedbe (master / detail)
dtPostDate	Datetime	Datum zapisa v podatkovno bazo
bitGlobalConnectionString	Int	Zastavica ali je za izvedbo potrebno uporabiti globalni povezovalni niz

Tabela **SystemBillingType** predstavlja lestvice, za katere so definirane poizvedbe. Na podlagi identifikatorja lestvice, tipa sistema in količine uporabe, ki jo dobimo s poizvedbo lahko ugotovimo za kateri artikel gre in tako izvemo ceno uporabe storitve.

SystemBillingType	Tip	Opis polja
nSystemBillingTypeID	int	Identifikator postavke (lestvice)
nSystemTypeID	int	Identifikator tipa sistema
vcName	varchar(500)	Ime postavke (lestvice)
vcUnitOfMeasure	varchar(100)	Merska enota
dtPostDate	datetime	Datum zapisa v podatkovno bazo
nPrepareDataQueryID	int	Identifikator poizvedbe

4.2.3 Podatkovno skladišče

Podatkovno skladišče je del podatkovne baze, ki hrani podatke za prikaz obračunov. Entitetni tip **BillingInvoice** predstavlja posamezen obračun. Ta ima več postavk (**BillingData**), za katere so zapisane podrobnosti uporabe (**BillingDataDetail**).



Slika 4 Model podatkovnega skladišča

V tabelo **BillingInvoice** se shranjujejo zgenerirani obračuni. Posamezen objekt te tabele predstavlja končni obračun naročnika. Ta vsebuje postavke, ki so shranjene v povezani tabeli BillingData.

BillingInvoice	Tip	Opis polja
nBillingInvoiceID	Int	Identifikator obračuna
nInvoiceSpecificationID	Int	Identifikator specifikacije obračuna
nCustomerID	Int	Identifikator naročnika
nPeriod	Int	Obdobje
dtPeriodStart	Datetime	Začetni datum obdobja

dtPeriodEnd	Datetime	Končni datum obdobja
dtPostDate	Datetime	Datum zapisa v podatkovno bazo

Tabela **BillingData** hrani postavke, ki so vezane na obračune (BillingInvoice).

BillingData	Tip	Opis polja
nBillingDataID	int	Identifikator postavke
nBillingDataGUID	uniqueidentifier	Globalni identifikator postavke
nPriceListID	int	Identifikator artikla
nCustomerID	int	Identifikator naročnika
nSystemID	int	Identifikator sistema
nBillingInvoiceID	int	Identifikator obračuna
vcPriceListIDExt	varchar(100)	Zunanji ID artikla
vcName	varchar(500)	Ime postavke
vcDescription	varchar(1000)	Opis
vcUnitOfMeasure	varchar(100)	Merska enota
nPackageCount	decimal(38, 0)	Količina v paketu
nItemType	int	Tip
nPrice	decimal(16, 9)	Cena ene enote
nDiscountRate	decimal(16, 9)	Popust
vcCurrency	varchar(100)	Valuta
nPeriod	int	Začetek obdobja v int32
dtPostDate	datetime	Datum zapisa v podatkovno bazo
nQuantity	decimal(34, 4)	Količina
dtPeriodStartDate	datetime	Datum začetka obdobja
dtPeriodEndDate	datetime	Datum konca obdobja
nQuantityCount	decimal(19, 6)	Količina za ceno
nAmount	decimal(16, 9)	Znesek

Podrobnosti uporabe storitve (npr. ime aktivnega uporabnika brezpaperja.si) se zapišejo v tabelo **BillingDataDetail**, ki je s tujim ključem nBillingDataID vezana na entitetni tip BillingData.

BillingDataDetail	Tip	Opis polja
nBillingDataDetailID	int	Identifikator podrobnosti
nBillingDataID	int	Identifikator postavke
vcName	varchar(500)	Ime podrobnosti
nQuantity	decimal(34, 4)	Količina
dtPostDate	Datetime	Datum zapisa v podatkovno bazo
vcCostCenter	varchar(100)	Stroškovno mesto
vcUnitOfMeasure	varchar(100)	Merska enota
nPrice	decimal(16, 9)	Cena
nDiscountRate	decimal(16, 9)	Popust
nAmount	decimal(16, 9)	Cena s popustom

4.3 Administracijski vmesnik

4.3.1 Delovanje

Administracijski vmesnik je realiziran s Caliburn. Micro ogrodjem za izdelavo WPF / MVVM aplikacij. Caliburn.Micro mi je močno pomagal pri povezovanju ViewModela z Viewom, prikazovanju posameznih Viewov in pošiljanju sporočil med ViewModeli. V ozadju vmesnika za interakcijo s podatki v bazi skrbi razred DataLayer. DataLayer s podatkovno bazo manipulira prek razreda DbContext, ki je del ogrodja Entity Framework. DbContext omogoča poizvedovanje po podatkovni bazi in spremlja spremembe na objektih podatkovne baze, ki jih nato skupaj vnese v podatkovno bazo. V tem razredu se tudi lovijo in obravnavajo napake oz. izjeme, ki se zgodijo v interakciji s podatkovno bazo. Te napake se z gradnikom MessageBox uporabniku izpišejo na zaslon.

Administracijski vmesnik je zasnovan iz več pogledov, ki jih prikazujemo z ogrodjem Caliburn.Micro. Tehnologija MVVM nam omogoča večkratno uporabo istega pogleda, ki se mu spreminja le vsebina. Glavni pogled, ki omogoča pregled vseh zapisov nekega tipa, je ItemView. V njem je osrednji element ListView. Tega vmesnik napolni z objekti, ki smo jih izbrali v navigacijskem meniju (Naročniki, Poizvedbe, Računi, ...). Vsaka vrstica v preglednici je nov objekt oz. instanca pogleda. Tako je za vsakega šifranta narejen nov pogled zapisa v preglednici, ki ima prilagojena polja izbranega šifranta. Ti pogledi imajo ime sestavljeno iz predpone »Individual«, imena entitetnega tipa in končnice »View« (npr. IndividualCustomerView). V ViewModelih teh pogledov se vsi tuji ključi posodobijo z imenom entitete v tuji tabeli. Za dodajanje in urejanje šifrantov pa so implementirani pogledi,

ki imajo ime sestavljeno iz predpone »Add«, imena entitetnega tipa in končnice »View« (npr. AddCustomerView).

Kot rečeno, se vse operacije urejanja podatkovne baze izvajajo v razredu DataLayer. Dodajanje entitete je realizirano z metodo AddItem, ki ji v parametru podamo entitetni tip za vnos. Objekte brišemo z metodo DeleteItem, ki v podatkovni bazi poišče podani element in ga odstrani iz tabele. Urejanje elementov je realizirano tako, da se za urejeni objekt poišče originalen element v podatkovni bazi in nato vanj prepíše vse lastnosti iz modificiranega. Tako DataContext v objektu DataLayer spozna spremembe in jih lahko shrani.

4.3.1.1. Optimizacija poizvedb administracijskega vmesnika

V administracijskem vmesniku se v ogrodju Entity Framework večkrat izvedejo strukturo enake poizvedbe. V takih primerih lahko močno izboljšamo zmogljivost tako, da te poizvedbe prevedemo (ang. compile) le enkrat nato pa jih uporabimo večkrat z različnimi parametri. Kljub temu, da ogrodje .NET Framework od verzije 4.5 podpira avtomatsko pomnjenje LINQ poizvedb, smo z vnaprejšnjim prevajanjem poizvedb dosegli opazno izboljšavo.

V našem primeru smo princip »Compiled Queries« uporabili pri vnašanju imena naročnika in naziva paketne postavke v preglednici artiklov. Z uporabo »Compiled Queries« se je 485 artiklov posodobilo z imeni in prikazalo v preglednici v 0,25s. Brez tega pa je ista operacija potrebovala kar 1,89s.

Uporabljeni »compiled« poizvedbi

```
private static readonly Func<dataEntities, PriceList, Customer>
CompiledGetCustomer_PriceList =
System.Data.Objects.CompiledQuery.Compile((dataEntities context, PriceList pl) =>
(from c in context.Customers where c.nCustomerID == pl.nCustomerID select
c).FirstOrDefault());

private static readonly Func<dataEntities, PriceList, SystemBillingType>
CompiledGetSystemBillingType_PriceList =
System.Data.Objects.CompiledQuery.Compile((dataEntities context, PriceList pl) =>
(from c in context.SystemBillingTypes where c.nSystemBillingTypeID ==
pl.nSystemBillingTypeID select c).FirstOrDefault());
```

4.3.2 Osnovni pogled aplikacije

Glavni del aplikacija je velika preglednica z vsemi entitetami, ki smo jih izbrali v navigacijskem meniju v zgornjem delu aplikacije. Pod navigacijskim delom je na voljo tudi okno za filtriranje oz. iskanje po preglednici.

Zgodovina Naročnik Zajem Računi					
Zgenerirana poročila					
		Isči	Počisti	Število zapisov	30
Ime naročnika	Ime računa	Datum nastanka	Od datuma	Do datuma	Status
Ziga test 5	Testni račun popravljen	13.11.2012 8:03	1.10.2012	1.11.2012	OK
Ziga test 5	Testni račun popravljen	12.11.2012 8:11	1.10.2012	1.11.2012	OK
Ziga test 5	Testni račun popravljen	7.11.2012 8:00	1.10.2012	1.11.2012	OK
Ziga test 5	Testni račun popravljen	6.11.2012 8:06	1.10.2012	1.11.2012	OK
Ziga test 5	Testni račun popravljen	5.11.2012 8:01	1.10.2012	1.11.2012	OK
Ziga test 5	Testni račun popravljen	30.10.2012 8:03	1.9.2012	1.10.2012	OK
Ziga test 5	Testni račun popravljen	29.10.2012 8:00	1.9.2012	1.10.2012	OK
Ziga test 5	Testni račun popravljen	25.10.2012 8:06	1.9.2012	1.10.2012	OK
Ziga test 5	Testni račun popravljen	24.10.2012 8:03	1.9.2012	1.10.2012	OK
Ziga test 5	Testni račun popravljen	23.10.2012 8:00	1.9.2012	1.10.2012	OK
Ziga test 5	Testni račun popravljen	22.10.2012 8:08	1.9.2012	1.10.2012	OK
Ziga test 5	Testni račun popravljen	18.10.2012 10:49	1.9.2012	1.10.2012	OK
Ziga test 5	Testni račun popravljen	18.10.2012 10:49	1.9.2012	1.10.2012	OK
Ziga test 5	Test - Testni račun popravljen	18.10.2012 9:59	1.9.2012	1.10.2012	OK
Ziga test 5	Test - Testni račun popravljen	18.10.2012 9:58	1.9.2012	1.10.2012	OK
Ziga test 5	Testni račun popravljen	18.10.2012 8:56	1.9.2012	1.10.2012	OK
Ziga test 5	Test - Žigov račun	12.10.2012 13:58	1.9.2012	1.10.2012	OK
Ziga test 5	Test - Žigov račun	12.10.2012 13:53	1.9.2012	1.10.2012	(Postavka = 2002) ni bila najdena za Žiga testis(716759) ==> Žiga sistem
Ziga test 5	Žigov račun	12.10.2012 13:52	1.9.2012	1.10.2012	OK(Postavka = 2002) ni bila najdena za Žiga testis(716759) ==> Žiga sistem
Ziga test 5	Test - Žiga test 4 Arhiviraj si račun	20.9.2012 14:09	1.8.2012	1.9.2012	OK
Ziga test 5	Test - Žiga test 4 Arhiviraj si račun	20.9.2012 13:56	1.8.2012	1.9.2012	OK
Ziga test 5	Test - Žiga test 4 Arhiviraj si račun	20.9.2012 13:55	1.8.2012	1.9.2012	OK
Ziga test 5	Test - Žiga test 4 Arhiviraj si račun	20.9.2012 13:53	1.8.2012	1.9.2012	OK
Ziga test 5	Test - Žiga test 4 Arhiviraj si račun	20.9.2012 13:52	1.8.2012	1.9.2012	OK
Ziga test 5	Test - Žiga test 4 Arhiviraj si račun	20.9.2012 13:47	1.8.2012	1.9.2012	OK
Ziga test 5	Test - račun 23	20.9.2012 11:23	1.8.2012	1.9.2012	OK
Ziga test 5	Ziga test 4 Arhiviraj si račun	20.9.2012 10:38	1.8.2012	1.9.2012	OK
Ziga test 5	Ziga test 4 Arhiviraj si račun	20.9.2012 10:14	1.8.2012	1.9.2012	OK
Ziga test 5	Ziga test 4 Arhiviraj si račun	20.9.2012 9:55	1.8.2012	1.9.2012	OK
Ziga test 5	racun 23	19.9.2012 14:28	1.8.2012	1.9.2012	OK

Slika 5 Osnovni pogled aplikacije

4.3.2.1. Navigacija

Navigacija po različnih nastavitvah Mikrocop Billing Systema je realizirana z menijem v zgornjem delu aplikacije. Šifranti so v navigacijskem meniju združeni v logično smiselne enote:

- Zgodovina – pregled generiranih obračunov s statusom uspešnosti generiranja
- Naročnik – šifranti vezani na naročnike
 - Naročniki
 - Skupine naročnikov
 - Artikli
 - Sistemi naročnikov
 - Sistemi
- Zajem
 - Poizvedbe
 - Paketne postavke

- Računi
 - Opravila
 - Računi

4.3.3 Dodajanje in urejanje šifrantov

Šifrante dodajamo in urejamo v istem pogledu. V takih pogledih je za vse tuje ključe gradnik ComboBox, ki prikazuje imena iz povezane tabele. Za dodajanje novih nad preglednico pritisnemo gumb **Dodaj** in odpre se nam nova prazna maska za vnašanje. Element uredimo z dvojnim klikom nanj v preglednici. Odpre se nam enaka maska kot pri dodajanju, le da je ta že izpolnjena s podatki izbranega elementa. Nov ali spremenjen element shranimo z gumbom **Shrani**, s pritiskom na **Prekliči** pa prekličemo operacijo dodajanja ali urejanja.

The screenshot shows a web application interface with a dark theme. At the top, there are four tabs: 'Zgodovina', 'Naročnik', 'Zajem', and 'Računi'. The 'Naročnik' tab is active, and the title 'Naročnik (dodaj / uredi)' is displayed. The form contains the following fields:

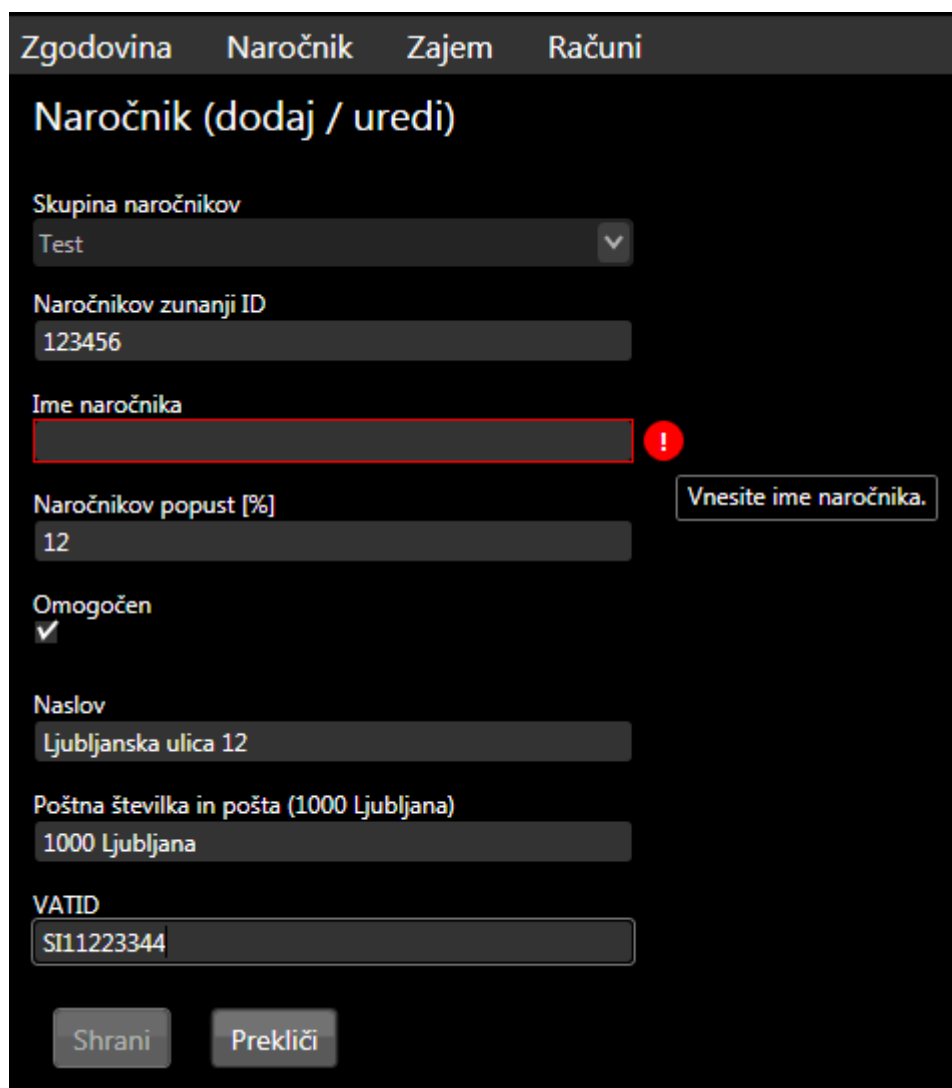
- Skupina naročnikov:** A dropdown menu with 'Test' selected.
- Naročnikov zunanji ID:** A text input field containing '123456'.
- Ime naročnika:** A text input field containing 'Testni naročnik'.
- Naročnikov popust [%]:** A text input field containing '12'.
- Omogočen:** A checkbox that is checked.
- Naslov:** A text input field containing 'Ljubljanska ulica 12'.
- Poštna številka in pošta (1000 Ljubljana):** A text input field containing '1000 Ljubljana'.
- VATID:** A text input field containing 'SI11223344'.

At the bottom of the form, there are two buttons: 'Shrani' (Save) and 'Prekliči' (Cancel).

Slika 6 Primer dodajanja novega ali urejanja naročnika

4.3.3.1. Validacija

Administracijskemu vmesniku smo dodali kontrole validacije. Dokler niso izpolnjena obvezna polja je gumb **Shrani** onemogočen, obenem pa so z rdečo označena polja, ki niso pravilno vnesena. Če se z miško sprehodimo čez polje napake, se nam v rdečem okenčku izpiše opis napake.



Slika 7 Nepravilno izpolnjena forma dodajanja novega naročnika

Validacija je realizirana v kontrolerju pogleda (ViewModel), ki za vsako polje na vnosni maski preveri prisotnost validacijskega pogoja in ob neustreznosti v pogled pošlje opis napake. Validacija obveznih ali občutljivih podatkov je prisotna pri dodajanju in urejanju katere koli entitete.

4.3.4 Brisanje šifrantov

Brisanje šifrantov je na voljo prek preglednice. Entitete, ki jih želimo zbrisati, označimo in pritisnemo gumb **Zbriši**.

Pri brisanju naročnikov se naročniki ne brišejo, temveč le onemogočijo.

4.3.5 Uporaba preglednice

V preglednici se izpiše omejeno število (1000) objektov. Če jih je v podatkovni bazi več kot toliko, se števec na vrhu okna obarva rdeče, poleg njega pa se zapiše še število vseh (npr. 1000 / 123000).

4.3.5.1. Filtriranje

Filtriranje se zgodi v razredu DataLayer z ukazi nad podatkovno bazo. S tem je optimizirana hitrost prikaza entitet, saj je filtriranje rezultatov v podatkovni bazi mnogo hitrejšo od filtriranja v aplikaciji. Pogled ItemView samo čaka na nov filtrirni vnos in nato posodobi preglednico. V posodabljanju pa se pri poizvedovanju po podatkovni bazi aplicira filter.

Omogočeno je filtriranje po imenu naročnika, če obstaja povezava na tabelo naročnikov, drugače se filtrira po imenu entitete (npr. ime računa). Iskalni algoritem ni občutljiv na velike črke in vrne tudi zapise, ki vsebujejo iskano besedo - se z njo ne ujamejo popolnoma. V iskalno polje nad preglednico vnesemo iskalni niz in pritisnemo tipko Enter ali gumb **Išči**. Z gumbom **Počisti** se odstrani iskalni niz in prikažejo vsi elementi.

4.3.5.1.1. »Cevovodno« filtriranje

Če v iskalni niz vnesemo več besed, ki jih ločimo z znakom "|" ali "OR", se bodo v preglednici izpisali vsi zapisi, ki ustrezajo katerikoli od teh besed.

Primer: filtriranje strank z nizom "test1 | test2 OR test3" nam prikaže stranke Test123, TEST2 in TesT333.

4.3.5.1.2. »Cevovodno« filtriranje artiklov

Za izpis splošnega cenika (ni vezan na nobenega naročnika) v iskalni niz vpišemo znak "_".

4.3.5.2. Dodajanje več artiklov hkrati

Zaradi lažjega dodajanja celotnega cenika naročnika smo v administracijski vmesnik dodali funkcijo dodajanja več artiklov hkrati.

V preglednici označimo več artiklov (vsaj dva), lahko si pomagamo z pritiskom tipke »Shift« in izbiro več artiklov hkrati, ki jih vse označimo z označitvijo le enega. Vsi označeni artikli se bodo po pritisku gumba **Dodaj cenik** izpisali v preglednici novega okna. Tu nato izberemo naročnika, ki mu želimo dodati cenik. Če naročnika ne izberemo, se bodo artikli v sistem dodali kot splošni (niso vezani na nobenega naročnika). Predno ga shranimo je možno tudi prilagajati višino popusta za posamezni artikel.

Označi	Tip sistema	Eksterni ID	Ime	Enota	Postavka	Cena	Popust[%]	Valuta	Na količino
<input type="checkbox"/>	Arhiviraj.si	180178	Hramba dok. v sistemu Arhiviraj.si(nad 2,5TB)	GB	Online storage in GB	1.8700	5	EUR	1.0000
<input type="checkbox"/>	Arhiviraj.si	180160	Hramba dok. v sistemu Arhiviraj.si(od 1 do 20 GB) / mesec	GB	Online storage in GB	8.0000	0	EUR	1.0000
<input type="checkbox"/>	Arhiviraj.si	180161	Hramba dok. v sistemu Arhiviraj.si(od 20 do 50 GB) / mesec	GB	Online storage in GB	7.2000	0	EUR	1.0000
<input type="checkbox"/>	Arhiviraj.si	180162	Hramba dok. v sistemu Arhiviraj.si(od 50 do 125 GB) / mesec	GB	Online storage in GB	6.4000	0	EUR	1.0000
<input type="checkbox"/>	Arhiviraj.si	180163	Hramba dok. v sistemu Arhiviraj.si(od 125 do 300 GB) / mesec	GB	Online storage in GB	5.6000	0	EUR	1.0000
<input type="checkbox"/>	Arhiviraj.si	180164	Hramba dok. v sistemu Arhiviraj.si(od 300 do 700 GB) / mesec	GB	Online storage in GB	4.8000	0	EUR	1.0000
<input type="checkbox"/>	Arhiviraj.si	180165	Hramba dok. v sistemu Arhiviraj.si(od 700 GB do 1,5 TB) / mesec	GB	Online storage in GB	4.0000	12	EUR	1.0000
<input type="checkbox"/>	Arhiviraj.si	180166	Hramba dok. v sistemu Arhiviraj.si(od 1,5 TB do 2,5 TB) / mesec	GB	Online storage in GB	3.2000	11	EUR	1.0000

Slika 8 Okno dodajanja več artiklov hkrati

4.4 Servisni modul

Servisni modul se neprestano izvaja kot Microsoft Windows servis. Celotna logika je implementirana v enem razredu (BillingService) in se izvaja v eni neskončni niti. Vse nastavitve za delovanje pridobi iz podatkovne baze, ki jo urejamo prek administracijskega vmesnika. Iz teh nastavitev modul zgenerira obračune, ki jih zapiše v isto podatkovno bazo.

4.4.1 Izvajanje opravil

Vsaki deset sekund za vsa opravila preveri ali je po urniku že čas za njihovo izvedbo. Testni obračuni se izvedejo takoj, kar se preverja s prisotnostjo ustrezne zastavice na opravilu (bIsTest).

V kontroli izvajanja po urniku se primerja čas zadnje izvedbe opravila s specificiranim. Opravilo se izvede, če čas zadnjega izvajanja ne ustreza šesturnem intervalu od specificiranega časa. S to kontrolo zagotovimo izvajanje vseh opravil, saj bi se drugače lahko z daljšim izvajanjem enega opravila istočasno opravilo izpustilo.

Ko pride opravilo na vrsto, se mu najprej nastavi čas zadnjega izvajanja zato, da se ne izvede večkrat. Nato se za vsako specifikacijo obračuna v opravilu definira obdobje. Obdobje je lahko zapisano v specifikacijah ali pa se samodejno definira na mesec pred generacijo. Če je za opravilo postavljena zastavica za ponovno generacijo, je potrebno zbrisati račune, ki so bili narejeni iz iste specifikacije računa in imajo isto obdobje.

Nato se preveri ali v podatkovnem skladišču že obstaja poročilo za to specifikacijo obračuna in istim obdobjem. Obračun se zgenerira le, če temu ni tako.

4.4.2 Postopek generiranja obračuna

Servisni modul za vse postavke na računu poišče poizvedbo s katero pridobimo podrobnosti uporabe postavke v določenem obdobju. Vse parametre, ki so definirani v poizvedbi, in povezovalni niz na podatkovno bazo, ki jo uporablja storitev, dobimo iz naročnikovega sistema (CustomerSystem). Ker so postavke lahko fiksne ali pa so vezane na lestvico (paketna postavka), jih je pri poizvedovanju količine uporabe potrebno obravnavati ločeno:

- Paketna postavka(poizvedba je definirana):
 - V začasno podatkovno strukturo se zapišejo vse podrobnosti uporabe produkta. Tako pridobimo količino. Z njo iz cenika naročnika oz. splošnega cenika razberemo artikel in s tem tudi znesek.

Če je na postavki tudi artikel že določen, se ugotovljena količina uporabe le deli s količino, za katero je vpisana cena artikla. Z delilnikom nato pomnožimo ceno artikla in tako izvemo polno ceno postavke.

Če je postavljena zastavica zaokroževanja količine navzgor (bCeiling), se v začasno podatkovno strukturo vpiše cena zaokrožena navzgor.

Če pa je omogočena zastavica fiksne količine, se namesto poizvedene količine uporabi količina, ki je določena na postavki.
 - V tabelo BillingDataDetail shranimo podrobnosti iz začasne podatkovne strukture. Npr. shranimo vse aktivne uporabnike v sistemu.
- Fiksna postavka(poizvedba ni definirana):
 - Iz specifikacije postavke (BillingSpecification) pridobimo ceno in količino uporabe. Npr. Module Integration – količina 1KOS – cena 100€.

Nato se postavki (BillingData) v podatkovnem skladišču vpišejo vsa potrebna polja iz specifikacije obračuna in artikla. Pri znesku se mora upoštevati tudi popust. Popust je lahko določen na posamezen artikel ali pa neposredno na stranko. Če je popust definiran na artiklu,

se uporabi tega, drugače pa se uporabi morebiten popust stranke (velja za vse storitve, na katere je naročena).

Po obdelavi vseh postavk na specifikaciji obračuna se v tabelo CustomerJobHistory vpiše status o generaciji. Če je bila generacija v celoti uspešna, se v status zapiše niz »OK«. Drugače se vpiše vanj opis napake, ki se je zgodila.

4.4.2.1. Testno generiranje obračuna

Za testiranje generiranja obračunov je postopek enak, le da se v podatkovno skladišče ne shranjuje zgeneriranega obračuna (BillingInvoice). Podatki o postavki se vpišejo v tabelo BillingDataTest, podrobnosti pa v BillingDataDetailTest. Po koncu testne generacije obračuna se njegove specifikacije (testno opravilo - Task, testna specifikacija obračuna - InvoiceSpecification) izbrišejo. S tem preprečimo večkratno izvedbo istega testnega obračuna. Specifikacije postavke se kaskadno brišejo na podatkovni bazi, ki mora na testni tabeli BillingDataTest imeti omogočeno opcijo »ON DELETE CASCADE«.

4.5 Windows reporting services

Windows Reporting Services je platforma, ki jo je potrebno namestiti na strežnik s podatkovno bazo, ki se uporablja v Mikrocop Billing System. Administracija prikazovanja poročil je najlažja kar prek spletnega vmesnika. Privzeto je spletni naslov poročil in administracije enak kot naslov podatkovne baze (na kateri so nameščeni Reporting Services) s pripono /reports (npr. <http://dataBase/reports>).

V pripravljen sistem je potrebno naložiti predloge poročil. Predloge poročil imajo določeno obliko prikaza in poizvedbe po podatkovnih bazah, kjer se nahaja podatkovno skladišče. V poizvedbah so vključeni parametri, ki jih uporabnik vnese v spletnem vmesniku in se uporabijo pri poizvedovanju po podatkovnem skladišču. Trenutno (2013) se uporablja poročilo, ki mu izberemo skupino naročnikov in obračunsko obdobje. To poročilo nam prikaže obračune storitev za vse naročnike, ki spadajo v izbrano skupino naročnikov.

Poročilo je narejeno v taki obliki, da lahko enostavno razširimo razdelek enega naročnika in odpre se nov razdelek z vsemi postavkami tega obračuna stranke. Tudi posamezen obračun in postavko v obračunu lahko razširimo in pregledamo vse podrobnosti uporabe.

Na slikah 9 in 10 sta prikazana primera obračuna za stranko *Žiga diploma*. V temno obarvanih vrsticah so posamezne postavke z vsemi podrobnostmi. Pri postavkah, ki imajo skupno količino uporabe storitve več od ena, lahko v belo obarvanih vrsticah pregledujemo podrobnosti uporabe.

Obracun po strankah s podrobnostimi

Naročnik	Artikel	Sistem	ID artikla	Skupna količina	Enota	Cena	Popust	Cena s popustom	Valuta	Naziv enote uporabe	Količina porabe
Žiga diploma											
	Modul Admin 180176	Arhiviraj.si - Testni diploma sistem	180176	1	KOS	250	0	250	EUR		
	Modul Integration 180175	Arhiviraj.si - Testni diploma sistem	180175	1	KOS	500	0	500	EUR		
	Uporabniki sistema Arhiviraj.si(od 1 do 10 uporabnikov) / mesec 180168	Arhiviraj.si - Testni diploma sistem	180168	3	KOS	12,5	0,2	30	EUR		3
										Better Barcelona	1
										Chad Araza	1
										Maryjane Arata	1
	Uvoz dok. v sistem Arh.si in paketno časovno žigosanje (od 1 do 10.000 dok) 450001	Arhiviraj.si - Testni diploma sistem	450001	8990	DOK	0,0125	0	112,375	EUR		8990
										NABAVA - Narocilnica	1610
										NABAVA - Predracun	261
										Prejeti racuni	7119

Slika 9 Primer končnega obračuna stranke za storitev InDoc RMS

Obracun po strankah s podrobnostimi

Naročnik	Artikel	Sistem	ID artikla	Skupna količina	Enota	Cena	Popust	Cena s popustom	Valuta	Naziv enote uporabe	Količina porabe
Žiga diploma											175
	Mesečno nadomestilo za uporabo sistema Brezpapirja.si 500052	Brezpapirja.si - Testni diploma bp.si sistem	500052	1	KOS	99	0	99	EUR		
	Št. začeti procesov (od 1 do 250) 500059	Brezpapirja.si - Testni diploma bp.si sistem	500059	157	KOS	0,4	0,05	59,66	EUR		157
										Test Workflow	131
										Test Workflow 2	26
	Uporabniki sistema Brezpapirja.si (od 6 do 25 uporabnikov) 500054	Brezpapirja.si - Testni diploma bp.si sistem	500054	18	KOS	8,5	0,05	145,35	EUR		18
										Armando Kolm	1
										Clarice Knowler	1
										Denice Nordlinger	1
										Derek	1
										Kreutzbender	
										Ella Pahnke	1
										Emilia Oxley	1
										Forest Orea	1
										Gisela Kosicki	1
										Jimmy Horsby	1
										Lamar McKibben	1
										Lucien Iurato	1
										Margarito	1
										Kornbau	
										Marshall Hutch	1
										Nelly Jakuboski	1
										Nicolas Khosravi	1
										Reginald Humes	1
										Sandra McAulay	1
										Val Osborne	1

Slika 10 Primer končnega obračuna stranke za storitev Brezpapirja.si

4.6 Uporaba programske rešitve

Cilj Mikrocopovega sistema za pripravo obračunov storitev je prikaz mesečnih obračunov po strankah. Za doseg tega cilja je najprej v administracijskem vmesniku potrebno vnesti podatke o strankah in storitvah, ki jih uporabljajo:

1. Osnovni objekt programske rešitve je naročnik. Za vnos naročnika moramo najprej vnesti skupino naročnikov.
2. Vnesi vse naročnike, ki še niso v podatkovni bazi, za katere želimo imeti obračune storitev.
3. Vnesi tipe sistemov – produktov, ki jih ponuja Mikrocop.
4. Vnesi paketne postavke.
5. Vnesi poizvedbe za ugotavljanje količine uporabe storitev. Poizvedbe bodo vnesli administratorji.
6. Za ugotavljanje cene uporabe storitve je potrebno vnesti cenike. Večina strank ima lastne, njim prilagojene, cenike. Zato vnesemo vse cenike odvisne od naročnikov ter tudi splošne cenike, ki niso vezani na stranko.
7. Vnesi naročnikove sisteme na katerih se uporabljajo storitve.

Ko so vsi šifranti po opisanem postopku uspešno vneseni, se lahko lotimo definiranja specifikacij obračunov in določanja kdaj se bodo generirali:

1. Vnesi specifikacije obračunov, ki definirajo kaj vse bo obsegal končni obračun.
2. Naredi opravilo, ki mu določiš urnik izvajanja in označiš vse obračune, ki naj jih zgenerira.

Z uspešnim vnosom vseh šifrantov in specifikacij za generiranje obračunov ima servisni modul dovolj informacij za pripravo obračunov storitev. Servisni modul se izvaja ves čas in je za uporabnika neviden. Uspešnost oz. neuspešnost generacije lahko vidimo v administracijskem vmesniku pod zavihkom **Zgodovina**.

Ko ugotovimo, da so bili vsi obračuni uspešno vneseni v podatkovno skladišče jih lahko izpišemo in izvozimo v različne formate prek spletnega brskalnika s pomočjo Microsoft Reporting Services preddefiniranih poročil.

4.7 Nadaljnji razvoj

Mikrocop Billing System ima še veliko prostora za nadaljnji razvoj. Ena glavnih funkcionalnosti, ki bi jih lahko v sistem vključili v prihodnje je verzioniranje cenikov strank. S to dodatno funkcionalnostjo bi pridobili zgodovinski pregled stroškov uporabe storitev posameznih naročnikov. Prav tako bi dosegli večjo transparentnost pri spreminjanju cen posameznih artiklov.

V prihodnje bi bilo smiselno v administracijski vmesnik dodati kontrolo za takojšnji vpogled v generirani obračun. Tako bi lahko ob vsaki spremembi nastavitve pregledali generirani obračun in po potrebi dodatno uredili nastavitve.

Obstoječa rešitev napake pri generiranju obračunov shranjuje le v podatkovno bazo. V prihodnje bi lahko razvili funkcionalnost obveščanja prek elektronske pošte. Po potrebi bi lahko tej funkcionalnosti dodali možnost izbire obveščanja o napakah in uspešnih generiranjih ali samo o napakah.

Za lažjo administracijo bi na administracijski vmesnik dodali kontrole za odpiranje uporabniške priročnika za uporabo. Poleg tega bi vmesniku lahko dodali gumb z dialogom, ki bi prikazal vse informacije o produktu.

Na nek način je nadaljnji razvoj Mikrocop Billing Systema odvisen tudi od drugih Mikrocop storitev. Praktično vsi novi oz. prilagojeni starejši produkti, ki podatke o uporabi vpisujejo v podatkovno bazo, se lahko samo z nastavitvami v administracijskem vmesniku dodajo v razvit sistem za pripravo obračunov storitev.

5 Zaključek

V diplomski nalogi je opisana programska rešitev Mikrocop Billing System – sistem za pripravo obračunov storitev. Sistem smo razvili v podjetju Mikrocop, ki razvit sistem uporablja že več kot eno leto.

Pri razvoju sistema smo se osredotočali na pravilno generiranje obračunov oz. polnjenje podatkovnega skladišča. Poleg tega smo veliko pozornosti posvetili enostavnosti uporabe administracijskega uporabniškega vmesnika. Dodatne funkcionalnosti kot so cevovodno filtriranje zapisov v preglednici, dodajanje več artiklov hkrati, validacija obveznih vnosnih polj in druge močno olajšajo administriranje.

Programska rešitev Mikrocop Billing System trenutno zadostuje potrebam interne uporabe v podjetju Mikrocop. Zato imajo možnosti za nadaljnji razvoj, ki smo jih tudi zapisali v zadnjem poglavju, nizko prioriteto. Vsekakor pa nadaljnji razvoj ni izključen, saj podjetje Mikrocop ponuja vedno večji spekter storitev. Tako bi se lahko tudi obračunavanje drugih tipov storitev vključilo v Mikrocop Billing System.

Literatura

1. V. Potočnik, Trženje storitev, Gospodarski vestnik, 2000
2. T. Glažar, Trženje računovodskih storitev v Sloveniji, Inštitut za računovodstvo, 2007
3. Domača spletna stran podjetja Mikrocop d.o.o.
Dostopna na: <http://mikrocop.com/> [julij 2013]
4. Free Sample Data for Testing
Dostopno na: <http://briandunning.com/sample-data/> [julij 2013]
5. Caliburn.Micro project webpage
Dostopno na: <http://caliburnmicro.codeplex.com/> [julij 2013]
6. Microsoft Reporting Services in Action
Dostopno na: <http://www.codeproject.com/Articles/9235/Microsoft-Reporting-Services-in-Action> [julij 2013]
7. Entity Framework Overview
Dostopno na: <http://msdn.microsoft.com/en-us/library/bb399567.aspx> [julij 2013]
8. Windows Presentation Foundation
Dostopno na: <http://msdn.microsoft.com/en-us/library/ms754130.aspx> [julij 2013]
9. LINQ (Language-Integrated Query)
Dostopno na: <http://msdn.microsoft.com/en-us/library/vstudio/bb397926.aspx> [julij 2013]
10. SQL Server Books Online
Dostopno na: <http://msdn.microsoft.com/enus/library/ms130214%28v=sql.105%29.aspx> [julij 2013]



Št. naloge: 01964 / 2013
Datum: 18.10.2013

Univerza v Ljubljani, Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Kandidat: **ŽIGA ŠINKO**

Naslov: **RAZVOJ APLIKACIJE ZA OBRAČUN STORITEV**
THE DEVELOPMENT OF APPLICATION FOR BILLING OF SERVICES

Vrsta naloge: Diplomsko delo univerzitetnega študija

Tematika naloge:

Obračun storitev je v primeru, ko storitve zaračunavamo veliko strankam po različnih cenovnih modelih, lahko zelo kompleksen sistem. Njegovo kompleksnost praviloma povečujejo še različni urniki obračuna ter specifične zahteve strank. Zasnujte in izdelajte načrt za aplikacijo za obračun storitev elektronskega arhiviranja dokumentov. Pri tem upoštevajte vse zahteve naročnika in opredeljena poslovna pravila za naslednja področja: cenovni modeli, specifikacije storitev za obračun, urniki obračunov ter analize in pregledi obračuna. Aplikacijo razvijte na Microsoft platformi in pri tem uporabite tudi Windows Reporting Services.

Mentor:

doc. dr. Rok Rupnik



Dekan:

prof. dr. Nikolaj Zimic